

Это очень короткий вводный курс в Систему Описания Ресурсов (RDF). Чтобы получить об RDF более подробные сведения, обратитесь к статье [«RDF в деталях»](#) (англ), на которой и была основана эта страница.

RDF — это способ представления знаний в децентрализованном мире; это основная технология Семантического Веба, который позволит компьютерным программам пользоваться всей структурированной информацией, распределённой по узлам Веба. Чтобы не останавливаться на этом в дальнейшем, сразу оговорим, что RDF — это не XML-формат в строгом смысле слова; это больше, чем просто метаданные; это имеет крайне мало отношения к RSS; и это не такой уж и сложно, как вам кажется.

## Общий план

RDF — это универсальный способ разложения любых знаний на маленькие кусочки. Он задаёт определённые правила касательно семантики, т.е. смысла этих кусочков. Идея состоит в том, чтобы одним простым способом можно было бы описать любой факт, притом в таком структурированном виде, чтобы его могли обрабатывать компьютерные программы. Вот пример RDF:

```
@prefix : <http://www.example.org/> .  
:john    a           :Person .  
:john    :hasMother  :susan .  
:john    :hasFather  :richard .  
:richard :hasBrother :luke .
```

Смысл здесь очевиден. Дальше мы рассмотрим этот пример подробнее.

Если вы знаете XML, то вот краткое сравнение. Как RDF, так и XML — простые и универсальные технологии. XML может использоваться для представления более абстрактных вещей, чем тот текст с угловыми скобками, которым он записан: им может задаваться DOM какой-либо древовидной структуры. Точно так же, RDF — это не просто формат записи; это формат представления информации, организованной в виде графов или сетей. Вы можете записывать RDF в виде XML; многие так и поступают. Тогда предыдущий пример выглядел бы так:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ns="http://www.example.org/#">
  <ns:Person rdf:about="http://www.example.org/#john">
    <ns:hasMother rdf:resource="http://www.example.org/#susan" />
    <ns:hasFather>
      <rdf:Description rdf:about="http://www.example.org/#richard">
        <ns:hasBrother rdf:resource="http://www.example.org/#luke" />
      </rdf:Description>
    </ns:hasFather>
  </ns:Person>
</rdf:RDF>
```

Но использовать XML необязательно — я, например, его не использую. Формат N3, показанный первым, ничуть не хуже.

То, в чём [RDF](#) сильнее всего отличается от XML и других технологий — это то, что RDF предназначен для представления знаний в распределённом мире. Другими словами, для RDF

особенно важен смысл. Всё, с чем работает RDF, имеет определённый смысл, — ссылается на какой-то конкретный объект, или на абстрактное понятие, или на некий факт. Стандарты, основанные на RDF, описывают логические выводы, связывающие эти факты, и указывают, как можно найти сами факты в огромной базе данных всех знаний, представленных в RDF.

RDF подходит для работы с распределёнными знаниями потому, что приложения могут собирать воедино RDF-файлы, размещённые в Интернете разными людьми, и с лёгкостью узнавать из собранного документа даже те новые вещи, которых не было ни в одной из его частей. В RDF предусматривается два процесса, благодаря которым это осуществляется, — во-первых, объединяются документы, использующие общие языки, и во-вторых, допускается использовать любые языки в каждом из документов. Эта гибкость — одна из отличительных черт RDF.

Рассмотрим второй пример RDF-документа:

```
@prefix : <http://www.example.org/> .  
:richard :hasSister :rebecca  
{ ?a :hasFather ?b . ?b :hasSister ?c . } => { ?a :hasAunt ?c }  
.
```

В этом RDF-документе определяется смысл отношения «быть тёткой» — оно выражается через два других отношения. Легко представить себе, как приложение, объединяющее этот документ с документом из первого примера, сможет понять из них, что `:rebecca` — тётка `:john-a`. Это возможно потому, что имена сущностей глобальные; если в одном документе используются `:john` и `:hasFather`, то приложения подразумевают, что они имеют тот же самый смысл и во всех остальных документах с тем же значением `@prefix`.

Так зачем же нужен RDF? Вот список областей его применимости, приведённый Ричардом Сиганиаком в [списке рассылки W3C Semantic Web](#):

- Вам нужно объединять данные из различных источников, не прибегая к созданию специализированных программ.
- Вам нужно дать другим доступ к вашим данным.
- Вам нужно децентрализовать ваши данные так, чтобы ими всеми не «владел» кто-то один.
- Вам нужно сделать что-то особенное с большими объёмами данных — вводить, извлекать, просматривать, анализировать, выполнять поиск, и т.д. Вы хотите создать (либо использовать готовый) универсальный инструмент, который бы позволял вам всё это делать, основываясь на модели данных RDF (имеющей то преимущество, что она не привязана к закрытым технологиям хранения и представления данных — в отличие от диалектов СУБД).

## Определение RDF

RDF можно определить как совокупность трёх простых правил:

1. Факт выражается тройкой вида (Подлежащее, Сказуемое, Дополнение) — похожей на простое предложение на естественном языке.
2. Подлежащие, сказуемые и дополнения — это имена сущностей реального мира, конкретных или абстрактных. Имя может быть 1) глобальным, ссылающимся на одну и ту же сущность во всех RDF-документах, где оно используется, либо 2) локальным, и тогда на сущность, на которую ссылается это имя, нельзя сослаться из-за пределов RDF-документа непосредственно.
3. Дополнения, кроме этого, могут быть текстовыми строками — «литералами».

Факты вы уже видели. Каждая из этих строк — факт:

```
:john      a          :Person .
:john      :hasMother :susan .
...
```

Имена бывают двух типов. Глобальные имена, имеющие всюду один и тот же смысл, всегда имеют вид URI (Универсальных Идентификаторов Ресурсов). Синтаксис (формат) URI может быть таким же, как у адреса веб-сайта; вы можете встретить RDF-файлы, где используются URI наподобие `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`; такие URI задают глобальные имена для каких-нибудь сущностей. Тот факт, что это похоже по виду на веб-адрес — чистое совпадение. Может быть, действительно существует веб-сайт с таким адресом, а может быть, его не существует, — это не имеет значения. Есть и другие типы URI, не начинающиеся с `http:`. URN — один из подтипов URI, и он используется, например, для идентификации книг по их ISBN-номеру — `urn:isbn:0143034650` и т.п. Другой универсальный тип URI — это TAG; пример такого URI — `tag:govtrack.us,2005:congress/senators/frist`. URI используются для глобальных имён потому, что они позволяют разбить пространство всех возможных имён на блоки, за которыми закреплены владельцы. Так, если URI начинается на `http://www.rdfabout.net/`, значит за него отвечаю я.

Это очень важный принцип, и поэтому я его повторяю: какого бы типа они ни были, URI используются в RDF-документах только в качестве имён сущностей, и больше ни для чего. Забудьте, что эти URI имеют какое-то отношение к Вебу.

Так как URI могут быть довольно длинными, то в форматах, используемых для представления RDF, они обычно сокращаются, используя перенятый из XML механизм «пространств имён». Именно поэтому в именах `:john`, `:hasMother` и других сущностей в приведённых примерах стоят двоеточия — они означают, что используются сокращённые имена. В наших примерах им соответствовали полные имена `http://www.example.org/#john`, `http://www.example.org/#hasMother` и т.д.

При записи URI обычно заключаются в угловые скобки, чтобы их можно было отличить от сокращённых имён.

Литералы позволяют включать в RDF текст. Они используются особенно интенсивно, когда с помощью RDF представляются метаданные:

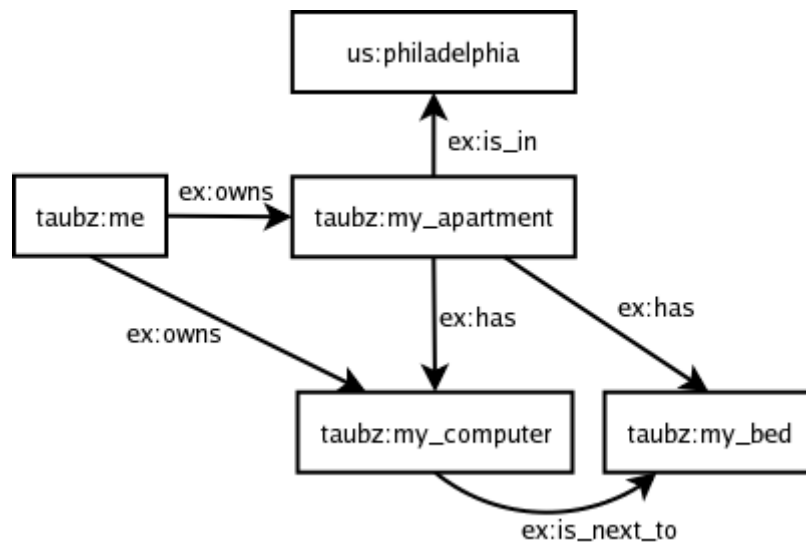
```
<http://www.rdfabout.net/> a :Website .
<http://www.rdfabout.net/> dc:title "rdf:about" .
<http://www.rdfabout.net/> dc:description "A website about
Resource Description Framework." .
```

Вот и все основы RDF.

## RDF в виде графа

Есть два взаимодополняющих способа рассматривать информацию, представленную в RDF. Первый способ — считать её набором утверждений, как в примерах выше: каждое утверждение представляет собой факт. Второй способ — считать её графом.

Граф — это, в общем, то же самое, что сеть. Граф состоит из узлов, соединённых рёбрами. Например, в Интернете узлы — это компьютеры, а рёбра — соединяющие их сетевые шнуры. В RDF узлы — это имена (но не сами сущности), а рёбра — утверждения. Например:



Здесь каждая стрелка (ребро) — это RDF-утверждение: имя у начала стрелки — это подлежащее утверждения, имя у конца стрелки — его дополнение, и имя у самой стрелки — сказуемое. Когда RDF представлен в виде графа, он содержит всю ту же информацию, что и выписанный в виде троек-утверждений; но графическое представление позволяет человеку легче увидеть структуру описываемых данных.

## Практический пример

Так чем же полезен RDF? Тем, что это самая подходящая технология, когда нужно свести распределённую информацию в общую систему.

Вот пример ситуации, когда информация была бы распределённой: база данных о продуктах различных поставщиков и оценках этих продуктов различными экспертами. Ни один поставщик не согласится взять на себя ответственность за поддержание общей центральной базы данных, в том числе и потому, что она содержала бы информацию о его конкурентах, и возможно, отрицательные оценки его собственной продукции. У экспертов же может не хватать ресурсов для постоянного поддержания в общей базе данных актуальной информации.

RDF бы подошёл в этом проекте особенно хорошо. Каждый поставщик и каждый эксперт бы размещали на собственном веб-сервере RDF-файл. Поставщики выбирают для своих продуктов URI, и эксперты используют эти URI, когда публикуют свои оценки. Тогда поставщики не вынуждены выбирать общую схему наименования для своих продуктов, и эксперты не привязаны к форматам данных, выбранным поставщиками. RDF позволяет и поставщикам, и экспертам использовать те инструменты, которые им удобнее, и никто не заставляет никого пользоваться каким-то определённым языком.

Такой вид имели бы RDF-файлы, которыми они обмениваются:

```

Поставщик 1:
vendor1:productX    dc:title      "Cool-O-Matic" .
vendor1:productX    retail:price   "$50.75" .
vendor1:productX    vendor1:partno "TTK583" .
vendor1:productY    dc:title      "Fluffertron" .
vendor1:productY    retail:price   "$26.50" .
vendor1:productY    vendor1:partno "AAL132" .

Поставщик 2:
vendor2:product1    dc:title      "Can Closer" .
vendor2:product2    dc:title      "Dust Unbuster" .

Эксперт 1:

```

```
vendor1:productX      dc:description "This product is good buy!"
```

```
.
```

#### Эксперт 2:

```
vendor2:product2  dc:description "Who needs something to unbust dust?  
A dust buster would be a better idea,  
and I wish they posted the price." .  
vendor2:product2  review:rating  review:Excellent .
```

Отдельный вопрос — как именно они обменивались бы этими файлами; его пока оставим в стороне. Как только приложение получает эти файлы, у него появляется достаточно информации, чтобы соотнести продукты, цены, оценки, и даже такую специфическую информацию, как `vendor1:partno` (шифр изделия). То, что вы должны понять из этого примера, — это насколько гибок RDF: он не накладывает практически никаких ограничений и всё же позволяет приложениям моментально соотносить распределённую информацию.

Поставщикам и экспертам не приходилось идти на компромиссы: они договорились использовать RDF, и всё. Им не приходилось договариваться о конкретном формате данных, или даже о конкретных URI. Что важнее всего, им не пришлось перечислять заранее все пункты, которые поставщик может включить в информацию о своём продукте, и в этой системе ни одного из экспертов невозможно лишить возможности публиковать свои оценки.

Кроме того, можно рассмотреть эту систему с точки зрения взаимной совместимости. Формат, используемый Поставщиком 1, полностью совместим с форматом, используемым всеми остальными, хотя они не вырабатывали общий формат целенаправленно. Когда в системе появится новое действующее лицо, которое захочет работать с информацией Поставщика 1, ему не потребуется реализовывать новый формат: всё что ему будет нужно — это выбрать те же подлежащие, сказуемые и дополнения, которые выбрал Поставщик 1.

## Заключение

Если вы думали, что RDF — это сложно, то надеюсь, теперь вы поняли, что это совсем не обязательно. RDF — это лёгкая, гибкая и необременительная технология. С ней становится лёгким моделирование знаний и объединение распределённых источников данных.

Чтобы получить об RDF более подробные сведения, переходите к статье [«RDF в деталях»](#) (англ).